nology in the learning process. However, the theoretical and practical study of the students' perception of information from printed and electronic sources have shown the effectiveness of perception of printed sources. For long-term use of computer sources in the process of education of students declining perception and attention, as well as the ability for abstract thinking and long-term memorization.
Keywords: computer technology for higher education, the efficiency of information perception of students, print and electronic sources of information.

УДК 512.541+512.542

РАЗРАБОТКА ИГР НА ЧЕРНО-БЕЛЫХ ПОЛЯХ С ДВУМЯ СХЕМАМИ ПРЕОБРАЗОВАНИЙ С ИСПОЛЬЗОВАНИЕМ VBA EXCEL

И.Н. Попов[1]

1 *popovivannik@yandex.ru*; Северный (Арктический) федеральный университет имени М.В. Ломоносова

Рассматривается вопрос о мотивации к изучению программирования. Решение задач о разложении элементов матричных групп через образующие с последующим представлением результатов в виде компьютерных игр может подтолкнуть к желанию или необходимости изучения VBA Excel.

Ключевые слова: программирование, VBA Excel.

Для решения прикладных и научно-исследовательских задач следует выбирать программное обеспечение (ПО), возможности которого более всего подходят для записи данных (входных и выходных) и реализации решения, исходя из стандартных (встроенных) команд и процедур. На практике встречается ситуация, что для решения задачи возникает необходимость дополнить стандартный набор пользовательским инструментарием, что приводит к использованию средств программирования выбранного ПО.

Мотивацией к изучению программирования могут являться задачи, реализующие теоретические абстрактные расчеты в виде чего-то конкретного, как одного из итогов исследования. Задачи такого типа появляются, в частности, из решения проблем абстрактной алгебры. Необходимость в умении программировать проявляется, с одной стороны, в нахождении прикладного характера теории, тем самым показывая ее важность, с другой стороны, в иллюстрации правильности полученных теоретических результатов конкретными примерами или нахождения презентабельных поясняющих примеров для демонстрации теорем теории.

В теории конечно-порожденных групп решается алгоритмическая проблема разложения, которая требует предложить алгоритм, по которому можно разложить любой ее элемент через образующие группы. В ряде случаев решение проблемы разложения имеет прикладной характер. Разработав алгоритм, нахождение самих разложений следует находить с использованием компьютерных программ из-за большого объема и сложности вычислений.

Одним из примеров конечно-порожденных групп являются аддитивные матричные группы с элементами, равными 0 и 1. Каждой матрице можно сопоставить прямоугольник, называемый полем, составленный из одинаковых квадратов двух

цветов: один цвет соответствует числу 1, другой — 0. Преобразование поля — правило, по которому за один раз можно перевернуть ряд квадратов поля обратной стороной. Образованному полю сопоставляется матрица, которая получается из матрицы исходного поля путем добавлением некоторой матрицы, соответствующей матрице-образующей группы [2].

Явно прослеживается идея представления полей и преобразований как элементы некоторой игры, целью которой является получения определенного поля, например, квадраты которого имеют заданную цветность [3]. В этом проявляется практическая значимость рассматриваемой теории. Программировать подобные игры можно в Excel, используя VBA [4]. На использование Excel подталкивает сходство строения матриц и представление данных в виде таблиц в этом ПО.

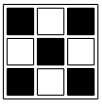
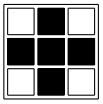
При программировании в Excel возможно сохранение значений переменных в ячейках листов, что позволяет не определять переменные как глобальные и получать доступ к значениям этих переменных из процедур, определенных в различных Excel-объектах и модулях. Отдельные листы могут содержать специально подготовленную информацию, которую не следует генерировать каждый раз при использовании пользовательских процедур.

Используя инструментарий встроенной системы VBA, на листе Excel можно создать понятный для пользователя интерфейс для проведения сеанса игры. Часть служебной информации может при этом храниться в ячейках листа, скрытая для игрока. В качестве помощи может быть представлен перечень преобразований для достижения цели игры. Для этого в коде программы должна быть часть, реализующая разложение матрицы поля в сумму матриц-образующих.

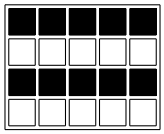
Рассмотрим следующую игру.

Предположим, что квадраты, одна сторона которых черная, вторая — белая, собраны в прямоугольник (поле игры) с 4 строками и 5 столбцами. Будем говорить, что дано поле. Правила игры: за один ход можно перевернуть обратной стороной квадраты, образующие квадрат с 3 строками и 3 столбцами, согласно одной из двух указанных схем преобразований. Естественно говорить, что поле преобразуется в поле. Цель игры заключается в том, чтобы за конечное число ходов преобразовать исходное поле в поле определенного вида или в поле, в котором все квадраты обращены вверх белой стороной (которое назовем пустым полем).

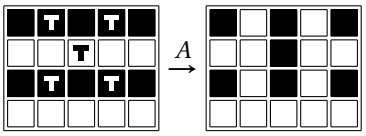
Например, схемы преобразований могут быть следующих «крестовых» видов:

A:  B: 

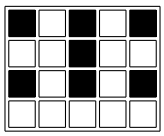
Схемы показывают следующее: на поле выбираются 9 квадратов, образующие квадрат 3 на 3, и те квадраты, которые закрашены черным на схеме, переворачиваются обратной стороной в выбранной части поля. Например, если поле имеет вид:



то выбрав схему A, получаем следующее преобразование данного поля:

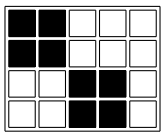


Стилизованная буква «Т» в клетках, образующих поле, указывает, какие квадраты именно переворачиваются. Получаем из исходного поля поле:

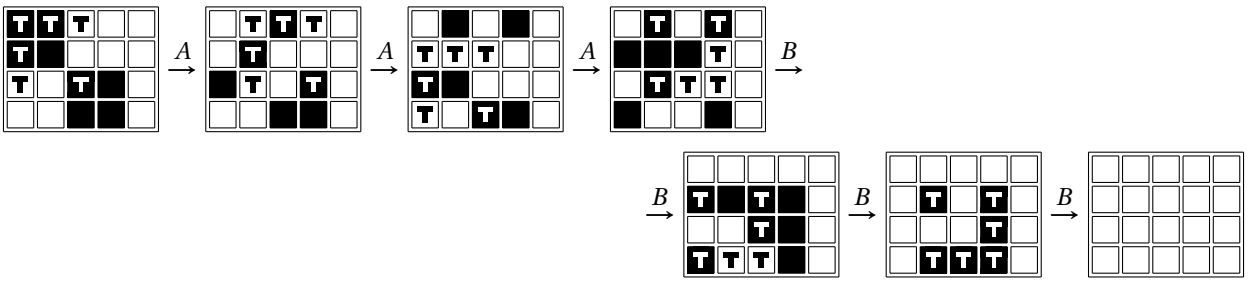


Пример. Рассмотрим схемы преобразований следующего вида:

Пусть дано поле:



С помощью указанных схем преобразований данное поле можно привести к пустому полю следующим образом:



Количество шагов преобразования равно 6.

Рассмотрим исследование софрмулированной игры с точки зрения математики и компьютерной реализации.

Каждому полю можно сопоставить матрицу размерности 4×5, следующим образом: черной клетке поля сопоставляется 1, белой — 0. С помощью схем преобразований порождаются 12 матриц размерности 4×5, по шесть матриц для каждой из схем преобразования. Например, по «крестовым» схемам преобразований порождаются следующие матрицы:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Обозначим матрицы символами $M_1,\ldots,M_{12}$. Тогда, если цель игры формулируется так, что поле следует привести к пустому полю, каждому полю сопоставляется матрица, равная сумме $\sum_{i=1}^{12}\varepsilon_i M_i$, где $\varepsilon_1,\ldots,\varepsilon_{12}$ принимают значения 0 или 1, где сложение подчиняется равенствам: $0+1=1+0=1$ и $0+0=1+1=0$. Отсюда следует то, что число различных полей с двумя схемами преобразований размерности 3×3 не превышает $2^{12}=4096$.

Матрицы $M_1,\ldots,M_{12}$ можно хранить на отдельном листе Excel-книги, скажем, на листе «CM», что способствует доступ к ним из любого модуля и Excel-объекта (например, подпрограмм, относящихся к листам книги). Сгенерировать такие матрицы на листе «CM» можно с помощью следующих команд:

```
For i = 1 To 24: For j = 1 To 10: Worksheets("CM").Cells(i, j) = 0: Next j, i
For k = 0 To 5: For i = 1 To 3: For j = 1 To 3
Worksheets("CM").Cells(4 * k + k \ 3 + i, k Mod 3 + j) = Cells(8+i, 9+j)
Next j, i, k
For k = 0 To 5: For i = 1 To 3: For j = 1 To 3
Worksheets("CM").Cells(4 * k + k \ 3 + i, k Mod 3 + j + 5) = Cells(8+i, 15+j)
Next j, i, k
```

Здесь считается, что левый верхний угол схемы преобразования A находится в 9 строке и 10 столбце, а схемы B — в 9 строке и 16 столбце. На листе «CM» в первых пяти столбцах будут расположены друг под другом матрицы, в которых подматрицей является матрица, соответствующая матрице схемы A, в столбцах с 6 по 10 — с подматрицей, соответствующей матрице схемы B.

В Excel можно запрограммировать действия при нажатии на ту или иную ячейку листа, не прибегая к использованию элемента управления формой «Кнопка». При нажатии на ячейку листа автоматически генерируются два числа, одно из которых соответствует номеру строки, в которой находится ячейка, второе — номер столбца. Для перехвата этих чисел в листинг программы следует добавить следующие строки:

```
Sub Worksheet_SelectionChange(ByVal cell As Range)
Dim numR, numC As Integer
numR = cell.Row: numC = cell.Column
End Sub
```

При программировании для обращения к ячейкам и изменяем их содержаний используются команды:

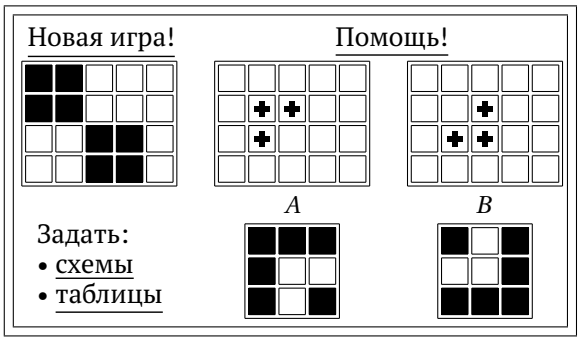
• Cells(R,C) (обращение к ячейке активного листа);

• Worksheets("Лист").Cells(R, C) (обращение к ячейке на листе с именем «Лист»), где R и C — номера строки и столбца ячейки на листе соответственно. Для установления и изменения форматов ячеек используются команды:

• Cells(R,C).Borders.LineStyle (формат границы (рамки) ячейки);

• Cells(R,C).Borders.Color (цвет границы (рамки) ячейки);
• Cells(R, C).Interior.Color (цвет поля ячейки);
• Cells(R, C).Font.ColorIndex (цвет текста в ячейке).

Игровая форма может иметь следующий вид:



На форме возможно активировать действие «Новая игра!», вызвать «Помощь!». Сверху формы расположены (слева направо) поле игры и две таблицы преобразований, внизу под которыми отображаются соответствующие схемы преобразований. Есть возможность изменять схемы и таблицы, активируя действия «Схемы» и «Таблицы».

Дадим пояснения к работе с формой.

• Поле игры

Основной частью формы является поле игры. Выбрав одну из предложенных схем *A* или *B* путем нажатия на любую из ее ячеек, нажимая ячейки поля игры, изменяются цвета ячеек этого поля согласно выбранной схеме преобразования. При этом считается, что нажатая ячейка поля игры соответствует центральной ячейке выбранной схеме преобразования. Поэтому активными ячейками поля игры являются только шесть центральных ячеек.

• Активация действия «Новая игра!»

Генерация новой игры ведется путем программного изменения ряда ячеек поля игры, выбирая случайным образом схемы преобразований. Сформированное поле игры игрок, выбирая схемы преобразования, должен привести к пустому полю, что является целью игры.

• Активация действия «Задать схемы»

Для того, чтобы изменить схемы преобразований, следует активировать поле «схемы». При этом слово «схемы» заменится на слово «ОК». При этом все ячейки полей схем станут белыми и, щелкая мышкой, составляются требуемые конфигурации схем. После этого следует щелкнуть на слово «ОК». Теперь в игре будут использоваться заданные новые схемы преобразований. При этом изменяются данные листа «СМ».

В частном случае схемы могут быть одинаковыми или одна из схем пустая.

• Активация действия «Задать таблицы»

При нажатии на слово «таблицы» оно заменяется словом «ОК». Каждой схеме *A* и *B* преобразований соответствует таблица преобразований. При нажатии на ячейки таблиц в них появляются стилизованные крестики. Каждый крестик означает позицию центра схемы преобразования. После нажатия на слово «ОК», во-первых, на

его месте появляется слово «таблицы», во-вторых, поле игры изменится согласно выбранным преобразованиям.

• Активация действия «Помощь!»

Главной целью вызова «Помощь!» является определение применения схем преобразований для преобразования поля к пустому полю. С точки зрения математики это означает то, что по матрице, соответствующей данному полю, находится коэффициенты $\varepsilon_1,\ldots,\varepsilon_{12}$, равные 0 или 1, определяется ее разложение $\sum_{i=1}^{12} \varepsilon_i \cdot M_i$ через матрицы-образующие $M_1,\ldots,M_{12}$.

Предварительно генерируются на отдельном листе «РМ» всевозможные наборы нулей и единиц, общее количество которых в каждом наборе равно 12, и которые упорядочены по числу единиц в наборе: от набора, состоящего только из нулей, до набора, состоящего из 12 единиц. Число таких наборов равно 4096. В общем случае число наборов из n нулей и единиц, в которых входит k единиц, $0 \le k \le n$, вычисляется по формуле $n!/(k!\cdot(n-k)!)$. Для определения всех требуемых наборов числа от 0 до 4095 представлены в двоичной системе счисления, и записаны в последовательных горизонтальных ячейках листа «РМ» (лист разложений матриц). Генерация наборов может быть осуществлена с помощью следующей программы:

```
Dim a, i, j, m, p As Integer
Dim S() As Integer
p = 12:  m = 4096
ReDim S(m) As Integer
For i = 1 To m: For j = 1 To p: Cells(i, j) = 0: Next j, i
For j = 0 To m - 1
   a = j: i = 0
   While a <> 0: i = i + 1: Cells(j + 1, p + 1 - i) = a Mod 2: a = a \ 2: Wend
   a = 0: For i = 1 To p: a = a + Cells(j + 1, i): Next i: S(j + 1) = a
Next j
For i = 1 To m: Cells(i, p + 1) = S(i): Next i
Call sortF(S(), m, p)
```

После запуска программы в ячейках в строках с 1 по 4096 (в столбца с 1 по 12) будут содержаться наборы нулей и единиц, в столбце 13 будет содержаться количество единиц в наборе. Производя сортировку наборов по 13 столбцу, получаем требуемое расположение наборов на листе «РМ». Сортировку можно провести средствами самого ПО Excel или используя метод сортировки всплытием Флойда сложности $O(n\log_2 n)$ [1], используя следующие подпрограммы:

```
Sub sortF(ByRef S() As Integer, ByVal m As Integer, ByVal p As Integer)
Dim i, k, vrem As Integer
For i = m \ 2 To 2 Step -1: Call MethodF(S, i, m, p): Next i
For k = m To 2 Step -1: Call MethodF(S, 1, k, p): Call swapline(k, 1, p):
vrem = S(k): S(k) = S(1): S(1) = vrem: Next k
End Sub

Sub MethodF(ByRef S() As Integer, ByVal i As Integer,
ByVal k As Integer, ByVal p As Integer)
Dim j, n, vrem As Integer
vrem = S(i): n = 2 * i
```

```
While n <= k
If n = k Then j = n Else If S(n) > S(n + 1) Then j = n Else j = n + 1
If S(j) > vrem Then Call swapline(i, j, p): S(i) = S(j): i = j: n = 2 * i
Else GoTo break
Wend
break: S(i) = vrem
End Sub

Sub swapline(ByVal u As Integer, ByVal v As Integer, ByVal p As Integer)
Dim vrem, k As Integer
For k = 1 To p
vrem = Cells(u, k): Cells(u, k) = Cells(v, k): Cells(v, k) = vrem
Next k
End Sub
```

Заметим, что после сортировки номер строки может не совпадать с его двоичным разложением.

Далее, берется набор с листа «РМ», начиная с самого верхнего, и формируется матрица SM, равная сумме $\sum_{i=1}^{12} \varepsilon_i \cdot M_i$. Если матрица SM для данного выбранного набора равна матрице, соответствующей матрице поля, то процесс определения коэффициентов прекращается. По найденному набору определяются матрицы разложения: первые шесть коэффициентов определяют, как применять схему A, остальные — схему B. При этом используется соответствие между наборами переменных и таблицами применений схем преобразований:

Набор переменных

ε_1 ε_2 ε_3 ε_4 ε_5 ε_6 ε_7 ε_8 ε_9 ε_{10} ε_{11} ε_{12}

Таблицы применения схем

	ε_1	ε_2	ε_3	
	ε_4	ε_5	ε_6	

	ε_7	ε_8	ε_9	
	ε_{10}	ε_{11}	ε_{12}	

Если некоторое ε равно 1, то в таблице появляется стилизованный «крестик», иначе клетка остается пустой.

Как видим, нахождение разложения ведется методом перебора. При этом разложение определяется с точки зрения минимизации как количества матриц-образующих, входящих в разложение (минимизации числа единиц в наборе коэффициентов), так и количества выполнения действий с матрицами.

Как отмечалось выше, каждая схема A и B определяет шесть матриц размерности 4×5. В листинге программы они определяются следующим образом:

```
Dim M(1 To 12, 1 To 4, 1 To 5) As Integer
```

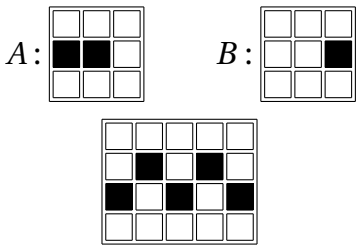
Тогда следующая часть программы определяет разложение матрицы $MF(i, j)$, соответствующая полю игры (точнее определяет номер строки n в листе «РМ»):

```
Dim SF(1 To 4, 1 To 5) As Integer
Dim i, j, n, k  As Integer
```

```
Dim bool As Boolean
bool = False: n = 0
While bool = False
   For i = 1 To 4: For j = 1 To 5: SM(i, j) = 0: Next j, i
   n = n + 1
      For k = 1 To 12: For i = 1 To 4: For j = 1 To 5
      SM(i, j) = (SM(i, j) + M(k, i, j) * Worksheets("PM").Cells(n, k)) Mod 2
      Next j, i, k
   bool = True
   For i = 1 To 4: For j = 1 To 5
      If MF(i, j) <> SM(i, j) Then bool = False
   Next j, i
Wend
```

По двенадцати элементам строки с номером n листа «PM» определяются ячейки матриц разложений.

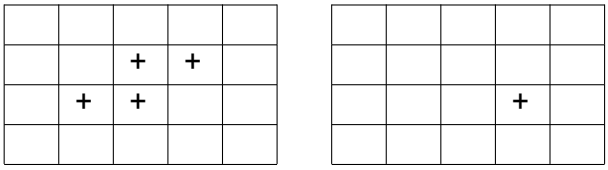
Например, пусть схемы преобразования A и B и поле игры имеют следующий вид соответственно:



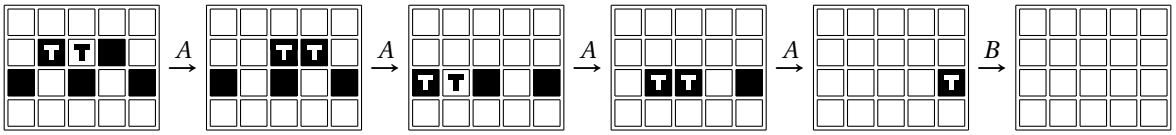
Тогда после работы программы в строке листа «PM» находится следующая последовательность из 12 нулей и единиц:

011110000001

По этому набору в таблицах применения схем определяются позиции, отмечаемые «+»:



Согласно этим таблицам получаем преобразования поля игры к пустому полю:



Всего использовано 5 преобразований. Отметим, что за меньшее число преобразовать данное поле к пустому нет возможности, хотя можно изменить порядок применения схем к достижению цели игры.

Рассмотренные в статье однотипные игры относятся к играм на черно-белых полях. Их всех объединяет то, что полем игры является прямоугольник, составленный из одинаковых квадратов с разноцветными сторонами, и преобразования заключаются в переворачивании квадратов поля. Каждая игра с точки зрения математики описывается некоторой матричной группой с элементами, равными 0 или 1. Игры отличаются лишь заданием преобразований. Разработка игр на черно-белых

полях требует знания не только знаний по программированию, но и знаний по таким математическим дисциплинам, как дискретная математика, абстрактная алгебра, теория матриц, теория векторных пространств. Учитывая возможности VBA Excel, часть которых упомянуты в статье, программирование игр на черно-белых полях можно вести как раз в ПО Excel. На это указывает способ ввода данных (например, активация ячеек листов и хранение матриц) и вывода данных (например, путем изменение цвета ячеек листов), возможность программирования и использование встроенных команд и процедур.

Литература

1. Иванов Б.Н. Дискретная математика. Алгоритмы и программы / Б.Н. Иванов. – М.: Лаборатория Базовых Знаний, 2001. – 288 с.

2. Попов И.Н. Группы *RC* и *RCD*: монография / И.Н. Попов. – Архангельск: КИРА, 2014. – 192 с.

3. Попов И.Н. Разбиения элементов множества на пары: генерация и применение / И.Н. Попов // Информационные технологии в образовании и науке – ИТОН 2016: материалы международной научно-практической конференции. – Казань: Изд-во Академии наук РТ, 2016. – С. 64–73.

4. Уокенбах Дж. Профессиональное программирование на VBA в Excel 2010 / Дж. Уокенбах. – М.: ООО «И.Д. Вильямс», 2012. – 944 с.

THE DEVELOPMENT OF GAMES ON THE BLACK AND WHITE FIELDS WITH THE TWO SCHEMES OF TRANSFORMATIONS USING VBA EXCEL

I.N. Popov

Addresses the issue of motivation to learn programming. The problem about decomposition of the elements of the matrix through forming groups with subsequent presentation of results in the form of computer games might push the desire or the need to learn VBA Excel.

Keywords: programming, VBA Excel.

УДК 378. 661: 378.47: 004.9

ФОРМИРОВАНИЕ ИКТ - КОМПЕТЕНЦИЙ - ПУТЬ К УСПЕШНОСТИ ОБУЧЕНИЯ СТУДЕНТОВ В МЕДИЦИНСКОМ ВУЗЕ

Н.М. Попова[1], Н.Г. Сабитова[2]

1 *kafedra-ozz@mail.ru*; ФГБОУ ВО “Ижевская государственная медицинская академи”
2 *sabitovang@mail.ru*; ФГБОУ ВО “Ижевская государственная медицинская академи”

В тезисе рассмотрены компетенции в области информационных и коммуникационных технологий (ИКТ-компетенции) в реализации дисциплины "Медицинская информатика необходимость разработки критериев, уровней и их содержание в сформированности ИКТ - компетенций на основе компетентностного подхода, оптимизацию и реализацию которого предлагается рассматривать в рамках компьютеризации и информатизации здравоохранения.

Ключевые слова: ФГОС-3, компьютеризация и информатизация здравоохранения, компетентностный подход, медицинская информатика, ИКТ - компетенции.